

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer implemented system that facilitates processing of a document, comprising a processor executing following components:
a host application that facilitates creation of the document; ~~and~~
a programming component that at least one of embeds code in the document and links code to another document such that the document can be run independently of the host application[[]] ; and
a data island of data comprised within the document accessible by a server and a client of the server such that the data island is modified on the server without having to start the host application on the server and contents of the data are synchronized with the document contents when the document runs inside the host application.

2. (Original) The system of claim 1, the document runs on a client and a server.

3. (Original) The system of claim 1, the programming component facilitates scaling by providing more features when the document is running on a client and fewer features when the document is running on a server.

4. (Original) The system of claim 1, the programming component separates document information into data content and view content.

5. (Original) The system of claim 4, the view content maps programmable names to generic API (Application Program Interface) objects, which objects are exposed as view controls that can be programmed against.

6. (Original) The system of claim 4, the data content acts indirectly against the view content via data binding.

7. (Currently Amended) The system of claim 1, the programming component generates ~~[[a]]~~ the data island in the document ~~that is accessible by a client and a server.~~

8. (Currently Amended) The system of claim 7, ~~the programming component automatically generates a~~ the data island in the document ~~that~~ conforms to a predetermined data schema and can be edited without the full host application running.

9. (Original) The system of claim 1, the programming component is event based such that the code runs according to an event that is related to a client or a server.

10. (Original) The system of claim 1 generates a runtime exception when a system error occurs.

11. (Original) The system of claim 1 controls permissions associated with the document according to whether the document is running on a client or a server.

12. (Original) The system of claim 1, the code includes data code portions of which are attributed to indicate if the corresponding data can be run on a client, a server, or both.

13. (Original) A computer according to the system of claim 1.

14. (Canceled)

15. (Currently Amended) A computer implemented system that facilitates processing of a document, comprising a processor executing the following components:
a host application that facilitates creation of the document; and
a data component that facilitates creation of a data island that is at least one of embedded in the document and linked to from another document such that the document can be run with only a subset of all components of the host application wherein the data island conforms to a data schema associated with the document, the data island is edited by running only a subset of components of the host application without having to start the host application and the data island is synchronized with document contents when the document is run inside the host application.

16. (Canceled)

17. (Canceled)

18. (Canceled)

19. (Original) The system of claim 15, further comprising a data model that is connected to the data island to work directly against data of the data island.

20. (Original) The system of claim 19, the data island is synchronized with document contents when the document is run inside the host application, and changes to the data model are moved into the document contents via a data binding mechanism.

21. (Original) The system of claim 15, data of the data island can be cached by marking the data using an attribute.

22. (Original) The system of claim 15, the document is one of an OLE structured document, an XML file, and a binary file that facilitates storing a persisted state of cached data, wherein if the document is a binary file, a reader/writer of the host application can be employed to insert the data island into the binary file and which reader/writer can be used to edit the data island.

23. (Original) The system of claim 22, when the OLE document is processed on either a client or a server, the cached data can be reconstituted out of the OLE document, manipulated, and changes to the cached data stored back into the OLE document.

24-40. (Canceled)

41. (New) A computer readable storage medium having stored thereon the following computer executable components:

- a host application that facilitates creation of the document;

- a programming component that at least one of embeds code in the document and links code to another document such that the document can be run independently of the host application; and

- a data island of data comprised within the document accessible by a server and a client of the server such that the data island is modified on the server without having to start the host application on the server and contents of the data are synchronized with the document contents when the document runs inside the host application.